OIPF

Release 2 Specification

Volume 5a – Web Standards TV Profile

A profile of HTML5 and other related web technologies for connected TVs

[V2.3] – [2014-01-24]

Open IPTV Forum

Open IPTV Forum

Postal address

Open IPTV Forum support office address
650 Route des Lucioles – Sophia Antipolis
Valbonne – FRANCE
Tel.: +33 4 92 94 43 83
Fax: +33 4 92 38 52 90

Internet
http://www.oipf.tv

Disclaimer

The Open IPTV Forum accepts no liability whatsoever for any use of this document.

Copyright Notification

Contents

Foreword

This Technical Specification (TS) has been produced by the Open IPTV Forum.

This specification provides multiple options for some features. The Open IPTV Forum Profiles specification complements the Release 2 specifications by defining the Open IPTV Forum implementation and deployment profiles.

1 Introduction

The Open IPTV Forum Release 2 Specification consists of ten volumes:

- Volume 1 – Overview [OIPF_OVIEW2],
- Volume 2 – Media Formats [OIPF_MEDIA2],
- Volume 2a – HTTP Adaptive Streaming [OIPF_HAS2],
- Volume 3 – Content Metadata [OIPF_META2],
- Volume 4 – Protocols [OIPF_PROT2],
- Volume 4a – Examples of Protocol Sequences [OIPF_PROTEX2],
- Volume 5 – Declarative Application Environment [OIPF_DAE2],
- Volume 5a – Web Standards TV Profile (the present volume),
- Volume 6 – Procedural Application Environment [OIPF_PAE2], and
- Volume 7 – Authentication, Content Protection and Service Protection [OIPF_CSP2].

This document is a profile of HTML, CSS and other related web technologies aimed at connected TV services and devices. Its goal is to describe a common profile that can be relied on by content and service providers and implemented by manufacturers. It does **not** describe extensions or modification to any of the referenced technologies but only tries to define a subset of web standards that are suitable and useful for TV deployments and at the same time stable enough to provide a good degree of confidence that real interoperability can be achieved. It may add clarifications and/or additional constraints where these are needed due to the nature of target deployment environment.

This document only describes a minimum subset of web technology that a Terminal compliant with this profile is required to support. This does not preclude terminals to support more technologies than the ones described in this profile.

This document will be updated over time, as maturity of different standards (now works in progress) increase and/or new web standards are defined.

This specification is aimed at TV terminals that implement a browser based application environment. Applications running in such environment are authored using a set of languages commonly referred to as "web technologies" or "web standards". This document lists the minimum set of languages that shall be supported by a Terminal conforming to this specification. A terminal may support more languages than the ones listed in this document.

To avoid fragmentation and enhance interoperability with other web technologies-based devices and eco-systems, this specification tries not to diverge from any of the referenced specifications it relies on. In some exceptional cases though, this specification may decide to intentionally diverge from the referenced specifications. Such differences will be **explicitly** noted throughout the document.

This document is organized as follows: the main body includes a list of references to specifications that this profile relies on and that are considered necessary to enable an enhanced user experience. Annex A contains instead a detailed list of which features for each specification are considered stable enough and therefore can be safely supported by terminals and used by application developers. Such tables will be updated in future revisions of this document as maturity of the various specifications evolve. Support tables are omitted for those specifications that are required to be fully supported.

2 References

2.1 Normative References

[CANVAS-2D-20121217]	Rik Cabanier; Eliot Graff; Jay Munro; Tom Wiltzius; Ian Hickson. *HTML Canvas 2D Context*. 17 December 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-2dcontext-20121217
[COOKIES]	Adam Barth. *HTTP State Management Mechanism*. April 2011. Internet Proposed Standard RFC 6265. URL: http://www.rfc-editor.org/rfc/rfc6265.txt
[CSS21-20110607]	Bert Bos et al. *Cascading Style Sheets, level 2 (CSS2) Specification*. 07 June 2011. W3C Recommendation. URL: http://www.w3.org/TR/2011/REC-CSS2-20110607/
[CSS3-ANIMATIONS-20130219]	Dean Jackson; David Hyatt; Chris Marrin; Sylvain Galineau; L. David Baron. *CSS Animations*. 19 February 2013. W3C Working Draft. URL: http://www.w3.org/TR/2013/WD-css3-animations-20130219/
[CSS3-BG-20120724]	Bert Bos; Elika J. Etemad; Brad Kemper. *CSS Backgrounds and Borders Module Level 3*. 24 July 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-css3-background-20120724/
[CSS3-CONDITIONAL-20130404]	David Baron. *CSS Conditional Rules Module Level 3*. 4 April 2013. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2013/CR-css3-conditional-20130404/
[CSS3-FONTS-20130212]	John Daggett. *CSS Fonts Module Level 3*. 12 February 2013. W3C Working Draft. URL: http://www.w3.org/TR/2013/WD-css3-fonts-20130212/
[CSS3-IMAGES-20120417]	Elika J. Etemad; Tab Atkins Jr.. *CSS Image Values and Replaced Content*. 17 April 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-css3-images-20120417
[CSS3-MEDIAQUERIES-20120619]	Håkon Wium Lie; Tantek Çelik; Daniel Glazman; Anne van Kesteren. *Media Queries*. 19 June 2012. W3C Recommendation. URL: http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/
[CSS3-TRANSFORMS-20120911]	Simon Fraser; Dean Jackson; David Hyatt; Chris Marrin; Edward O'Connor; Dirk Schulze; Aryeh Gregor. *CSS Transforms*. 11 September 2012. W3C Working Draft. URL: http://www.w3.org/TR/2012/WD-css3-transforms-20120911
[CSS3-TRANSITIONS-20130212]	Dean Jackson; David Hyatt; Chris Marrin; L. David Baron. *CSS Transitions*. 12 February 2013. W3C Working Draft. URL: http://www.w3.org/TR/2013/WD-css3-transitions-20130212/
[CSS3COL-20110412]	Håkon Wium Lie. *CSS3 module: Multi-column layout*. 12 April 2011. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2011/CR-css3-multicol-20110412/
[CSS3COLOR-20110607]	Tantek Çelik; Chris Lilley; L. David Baron. *CSS Color Module Level 3*. 07 June 2011. W3C Recommendation. URL: http://www.w3.org/TR/2011/REC-css3-color-20110607/
[CSS3TEXT-20121113]	Elika J. Etemad; Koji Ishii. *CSS Text Module Level 3*. 13 November 2012. W3C Working Draft. URL: http://www.w3.org/TR/2012/WD-css3-text-20121113/
[CSS3UI-20120117]	Tantek Çelik. *CSS3 Basic User Interface Module*. 17 January 2012. W3C Working Draft. URL: http://www.w3.org/TR/2012/WD-css3-ui-20120117/

[CSSOM-VIEW]	Anne van Kesteren. *CSSOM View Module*. 4 August 2011. W3C Working Draft. URL: http://www.w3.org/TR/2011/WD-cssom-view-20110804
[DOM-LEVEL-3-EVENTS-20120906]	Travis Leithead; Jacob Rossi et al. *Document Object Model (DOM) Level 3 Events*. 06 September 2012. Working Draft. URL: http://www.w3.org/TR/2012/WD-DOM-Level-3-Events-20120906/
[ECMA-262-51]	*ECMAScript Language Specification, Edition 5.1*. June 2011. URL: http://www.ecma-international.org/publications/standards/Ecma-262.htm
[FLEXBOX-20120918]	Tab Atkins Jr; Elika J. Etemad; Alex Mogilevsky. *CSS Flexible Box Layout Module*. 18 September 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-css3-flexbox-20120918/
[HTML5-20130806]	Robin Berjon; Steve Faulkner; Travis Leithead; Erika Doyle Navara; Edward O'Connor; Silvia Pfeiffer. HTML5. 6 August 2013. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2013/CR-html5-20130806/
[MEDIA-FRAGS-20120925]	Raphaël Troncy et al. *Media Fragments URI 1.0 (basic)*. 25 September 2012. W3C Recommendation. URL: http://www.w3.org/TR/2012/REC-media-frags-20120925/
[POSTMSG-20120501]	Ian Hickson. *HTML5 Web Messaging*. 01 May 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-webmessaging-20120501
[RFC2119]	S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels.* March 1997. Internet RFC 2119. URL: http://www.ietf.org/rfc/rfc2119.txt
[RFC6455]	I. Fette; A. Melnikov. *The WebSocket Protocol*. December 2011 including verified errata. URL: http://tools.ietf.org/html/rfc6455
[SELECTORS-LEVEL-3-20110929]	Daniel Glazman et al. *Selectors API Level 3*. 29 September 2011. W3C Recommendation. URL: http://www.w3.org/TR/2011/REC-css3-selectors-20110929/
[SSE-20121211]	Ian Hickson. *Server-Sent Events*. 11 December 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-eventsource-20121211/
[WEBSOCKETS-API-20120920]	I. Hickson. *The WebSocket API*. 20 September 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-websockets-20120920/
[WEBSTORAGE-20130730]	Ian Hickson. *Web Storage*. 30 July 2013. W3C Recommendation. URL: http://www.w3.org/TR/2013/REC-webstorage-20130730/
[WEBWORKERS-20120501]	Ian Hickson. *Web Workers*. 01 May 2012. W3C Candidate Recommendation. URL: http://www.w3.org/TR/2012/CR-workers-20120501
[WOFF-20121213]	Jonathan Kew; Tal Leming; Erik van Blokland. *WOFF File Format 1.0*. 13 December 2012. W3C Recommendation. URL: http://www.w3.org/TR/2012/REC-WOFF-20121213/
[XHR-20121206]	Julian Aubourg et al. *XMLHttpRequest.*. 6 December 2012. W3C Working Draft. URL: http://www.w3.org/TR/2012/WD-XMLHttpRequest-20121206/

2.2 Open IPTV Forum References

[OIPF_ARCH2]	Open IPTV Forum, “Functional Architecture - V2.3”, January 2014.
[OIPF_OVIEW2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 1 - Overview", V2.3, January 2014.
[OIPF_MEDIA2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 2 - Media Formats”, V2.3, January 2014.
[OIPF_HAS2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 2a - HTTP Adaptive Streaming”, V2.3, January 2014.
[OIPF_META2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 3 - Content Metadata”, V2.3, January 2014.
[OIPF_PROT2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 4 - Protocols”, V2.3, January 2014.
[OIPF_PROTEX2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 4a - Examples of IPTV Protocol Sequences”, V2.3, January 2014.
[OIPF_DAE2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 5 - Declarative Application Environment”, V2.3, January 2014.
[OIPF_DAE2_WEB]	Open IPTV Forum, “Release 2 Solution Specification, Volume 5a - Web Services TV Profile”, V2.3, January 2014.
[OIPF_PAE2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 6 - Procedural Application Environment”, V2.3, January 2014.
[OIPF_CSP2]	Open IPTV Forum, “Release 2 Solution Specification, Volume 7 - Authentication, Content Protection and Service Protection”, V2.3, January 2014.

2.3 Informative References

[DOM4]	Anne van Kesteren; Aryeh Gregor; Lachlan Hunt; Ms2ger. *DOM4*. 6 December 2012. W3C Working Draft. URL: http://www.w3.org/TR/dom/
[RFC3986]	T. Berners-Lee; R. Fielding; L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax (RFC 3986)*. January 2005. RFC. URL: http://www.ietf.org/rfc/rfc3986.txt

3 Conventions and Terminology

3.1 Conventions

The key words MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this specification are to be interpreted as described in [RFC2119].

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

This document references specifications that some times are still works in progress. All features marked as **"at risk"** in such referenced specification SHALL be considered as OPTIONAL to support unless this profile explicitly mandate support for them.

3.2 Terminology

3.2.1 Definitions

In addition to the Definitions provided in Volume 1, the following abbreviations are used in this Volume.

Term	**Definition**
Terminal	a device running an interactive user-agent (browser) conformant to this specification. This is equivalent to the the OITF block as defined in [OIPF_ARCH2].
Application	an interactive software authored using the set of languages defined by this profile and conformant with this profile.
Authoring tool	a software program used to edit applications.
MB	2^{20} bytes

3.2.2 Abbreviations

In addition to the Abbreviations provided in Volume 1, the following abbreviations are used in this Volume.

Acronym	**Definition**
API	Application Programming Interface
DOM	Document Object Model
CSS	Cascading Style Sheets
HTML	
URI	Uniform Resource Identifier
XHTML	
WOFF	Web Open Font Format

4 Markup

The HTML5 [HTML5-20130806] specification defines conformance requirements for user-agents and documents. Applications and authoring tools SHALL comply with conformance requirements for documents unless differently specified in annex A.1. Terminals SHALL comply with conformance requirements for user agents unless differently specified in annex A.1; in particular a terminal SHALL support the HTML syntax and the XHTML syntax for HTML documents as defined in [HTML5-20130806].

5 Style

Support for CSS as a whole is not required by HTML5, even though some features are defined in terms of specific CSS requirements. This section defines requirements for CSS and other style and/or graphic related technologies.

5.1 Basic Graphic

Terminals SHALL support CSS Basic User Interface [CSS3UI-20120117] as profiled in annex A.2.1.

Terminals SHALL support CSS 2.1 [CSS21-20110607]. Although the CSS 2.1 specification includes a `"tv"` media type, this has not been widely used in practice. Terminals MAY ignore the `"tv"` media type and just use the `"screen"` media type.

Terminals SHALL support CSS Color Module Level 3 [CSS3COLOR-20110607].

Terminals SHALL support CSS Image Values and Replaced Content [CSS3-IMAGES-20120417] as profiled in annex A.2.2.

Terminals SHALL support CSS Backgrounds and Borders [CSS3-BG-20120724] as profiled in annex A.2.3.

5.2 Device Adaptation, Layout and Processing

Terminals SHALL support CSS Selectors Level 3 [SELECTORS-LEVEL-3-20110929].

Terminals SHALL support CSS Media Queries [CSS3-MEDIAQUERIES-20120619]

Terminals SHALL support CSS Multi-column Layout [CSS3COL-20110412].

Terminals SHALL support CSS Flexible Box Layout [FLEXBOX-20120918].

Terminals SHOULD support CSS Conditional Rules Module Level 3 [CSS3-CONDITIONAL-20130404].

5.3 Text and Fonts

Terminals SHALL support the [CSS3-FONTS-20130212] as profiled in annex A.2.4.

Terminals SHALL support the Web Open Font Format (WOFF) [WOFF-20121213]. Applications can link to [WOFF-20121213] fonts via a `@font-face` rule ([CSS3-FONTS-20130212]).

NOTE

Note that WOFF packaged fonts may require a significant amount of space. See section 9 for some recommendations.

Terminals SHALL support CSS Text [CSS3TEXT-20121113] as profiled in annex A.2.5.

5.4 Advanced Graphic

Terminals SHALL support the CSS Transforms [CSS3-TRANSFORMS-20120911] as profiled in annex A.2.6.

Terminals SHALL support CSS Transitions [CSS3-TRANSITIONS-20130212] as profiled in annex A.2.7.

Terminals SHALL support CSS Animations [CSS3-ANIMATIONS-20130219].

Terminals SHALL support Canvas 2D [CANVAS-2D-20121217] as profiled in annex A.3.3.

6 Scripting

Scripts are small programs that can be embedded into applications. While defining features that rely on scripting, HTML5 does not mandate support for scripting for all user-agents. Furthermore scripting is defined using a syntax that in most cases is independent from the underlying scripting language. For such reasons, this specification has additional requirements as defined in this section.

6.1 ECMAScript

- Terminals SHALL support scripting as defined in [HTML5-20130806].
- Terminals SHALL support [ECMA-262-51] scripting language.

6.2 Event model

In addition to the support for the DOM 3 events specification that is required by HTML5 [HTML5-20130806], terminals SHALL support the focus and keyboard event types defined in section 5.2.2 and 5.2.5 respectively of [DOM-LEVEL-3-EVENTS-20120906].

NOTE

Note that [XHR-20121206] (referenced by this specification) also requires support for a subset of event related functionality defined in [DOM4], such as various exceptions and EventTarget.

In order to support legacy content, Terminals SHALL support also the legacy attributes `keyCode` and `charCode` as defined in Appendix B of [DOM-LEVEL-3-EVENTS-20120906].

Applications SHALL NOT rely on these attributes and SHALL use the new event model as defined in DOM3 Events. Values used for the `keyCode` property are implementation specific, but SHALL be exposed to the application through the constants below, defined on the `KeyboardEvent` interface. For each constant, an equivalent key value is given (as defined for `KeyboardEvent.key` in DOM3, see section 6.2.7.1 of [DOM-LEVEL-3-EVENTS-20120906]).

In order to support legacy content, these constants SHALL also be available via a `KeyEvent` interface. For example VK_OK can be accessed as `KeyEvent`.VK_OK.

NOTE

There is no requirement for Terminals to be able to generate all keycodes defined below.

Virtual Keycode Constants	Equivelant key value (in DOM3)
VK_UNDEFINED	‘Unidentified’
VK_CANCEL	‘Cancel’
VK_BACK_SPACE	‘Backspace’
VK_TAB	‘Tab’
VK_CLEAR	‘Clear’
VK_ENTER	‘Enter’
VK_SHIFT	‘Shift’
VK_CONTROL	‘Control’
VK_ALT	‘Alt’
VK_PAUSE	‘Pause’
VK_CAPS_LOCK	‘CapsLock’
VK_KANA	‘KanaMode’
VK_FINAL	‘FinalMode’
VK_KANJI	‘KanjiMode’
VK_ESCAPE	‘Esc’
VK_CONVERT	‘Convert’
VK_NONCONVERT	‘Nonconvert’
VK_ACCEPT	‘Accept’
VK_MODECHANGE	‘ModeChange’

VK_SPACE	space character, fix
VK_PAGE_UP	‘PageUp’
VK_PAGE_DOWN	‘PageDown’
VK_END	‘End’
VK_HOME	‘Home’
VK_LEFT	‘Left’
VK_UP	‘Up’
VK_RIGHT	‘Right’
VK_DOWN	‘Down’
VK_COMMA	‘,’
VK_PERIOD	‘.’
VK_SLASH	‘/’
VK_0	‘0’
VK_1	‘1’
VK_2	‘2’
VK_3	‘3’
VK_4	‘4’
VK_5	‘5’
VK_6	‘6’
VK_7	‘7’
VK_8	‘8’
VK_9	‘9’
VK_SEMICOLON	‘;’
VK_EQUALS	‘Equals’
VK_A	‘A’
VK_B	‘B’
VK_C	‘C’
VK_D	‘D’
VK_E	‘E’
VK_F	‘F’
VK_G	‘G’
VK_H	‘H’
VK_I	‘I’
VK_J	‘J’
VK_K	‘K’
VK_L	‘L’
VK_M	‘M’
VK_N	‘N’
VK_O	‘O’
VK_P	‘P’
VK_Q	‘Q’
VK_R	‘R’
VK_S	‘S’
VK_T	‘T’
VK_U	‘U’
VK_V	‘V’
VK_W	‘W’
VK_X	‘X’
VK_Y	‘Y’
VK_Z	‘Z’
VK_OPEN_BRACKET	‘[‘
VK_BACK_SLASH	‘\’
VK_CLOSE_BRACKET	‘]’
VK_NUMPAD0	See NOTE below
VK_NUMPAD1	See NOTE below
VK_NUMPAD2	See NOTE below
VK_NUMPAD3	See NOTE below
VK_NUMPAD4	See NOTE below
VK_NUMPAD5	See NOTE below

VK_NUMPAD6	See NOTE below
VK_NUMPAD7	See NOTE below
VK_NUMPAD8	See NOTE below
VK_NUMPAD9	See NOTE below
VK_MULTIPLY	'Multiply'
VK_ADD	‘Add’
VK_SEPARATER	‘Separator’
VK_SUBTRACT	‘Subtract’
VK_DECIMAL	‘Decimal’
VK_DIVIDE	‘Divide’
VK_F1	‘F1’
VK_F2	‘F2’
VK_F3	‘F3’
VK_F4	‘F4’
VK_F5	‘F5’
VK_F6	‘F6’
VK_F7	‘F7’
VK_F8	‘F8’
VK_F9	‘F9’
VK_F10	‘F10’
VK_F11	‘F11’
VK_F12	‘F12’
VK_DELETE	‘Del’
VK_NUM_LOCK	‘NumLock’
VK_SCROLL_LOCK	‘Scroll’
VK_PRINTSCREEN	‘PrintScreen’
VK_INSERT	‘Insert’
VK_HELP	‘Help’
VK_META	‘Meta’
VK_BACK_QUOTE	‘`’
VK_QUOTE	‘'’
VK_RED	‘Red’
VK_GREEN	‘Green’
VK_YELLOW	‘Yellow’
VK_BLUE	‘Blue’
VK_GREY	‘Grey’
VK_BROWN	‘Brown’
VK_POWER	‘Power’
VK_DIMMER	‘Dimmer’
VK_WINK	‘Wink’
VK_REWIND	‘MediaRewind’
VK_STOP	‘MediaStop’
VK_EJECT_TOGGLE	‘Eject’
VK_PLAY	‘MediaPlay’
VK_RECORD	‘MediaRecord’
VK_FAST_FWD	‘FastFwd’
VK_PLAY_SPEED_UP	‘PlaySpeedUp’
VK_PLAY_SPEED_DOWN	‘PlaySpeedDown’
VK_PLAY_SPEED_RESET	‘PlaySpeedReset’
VK_RECORD_SPEED_NEXT	‘RecordSpeedNext’
VK_GO_TO_START	‘MediaTrackStart’
VK_GO_TO_END	‘MediaTrackEnd’
VK_TRACK_PREV	‘MediaPreviousTrack’
VK_TRACK_NEXT	‘MediaNextTrack’
VK_RANDOM_TOGGLE	‘RandomToggle’
VK_CHANNEL_UP	‘ChannelUp’
VK_CHANNEL_DOWN	‘ChannelDown’
VK_STORE_FAVORITE_0	‘StoreFavorite0’
VK_STORE_FAVORITE_1	‘StoreFavorite1’

VK_STORE_FAVORITE_2	‘StoreFavorite2’
VK_STORE_FAVORITE_3	‘StoreFavorite3’
VK_RECALL_FAVORITE_0	‘RecallFavorite0’
VK_RECALL_FAVORITE_1	‘RecallFavorite1’
VK_RECALL_FAVORITE_2	‘RecallFavorite2’
VK_RECALL_FAVORITE_3	‘RecallFavorite3’
VK_CLEAR_FAVORITE_0	‘ClearFavorite0’
VK_CLEAR_FAVORITE_1	‘ClearFavorite1’
VK_CLEAR_FAVORITE_2	‘ClearFavorite2’
VK_CLEAR_FAVORITE_3	‘ClearFavorite3’
VK_SCAN_CHANNELS_TOGGLE	‘ScanChannelsToggle’
VK_PINP_TOGGLE	‘PinPToggle’
VK_SPLIT_SCREEN_TOGGLE	‘SplitScreenToggle’
VK_DISPLAY_SWAP	‘DisplaySwap’
VK_SCREEN_MODE_NEXT	‘ScreenModeNext’
VK_VIDEO_MODE_NEXT	‘VideoModeNext’
VK_VOLUME_UP	‘VolumeUp’
VK_VOLUME_DOWN	‘VolumeDown’
VK_MUTE	‘VolumeMute’
VK_SURROUND_MODE_NEXT	‘AudioSurroundModeNext’
VK_BALANCE_RIGHT	‘AudioBalanceRight’
VK_BALANCE_LEFT	‘AudioBalanceLeft’
VK_FADER_FRONT	‘AudioFaderFront’
VK_FADER_REAR	‘AudioFaderRear’
VK_BASS_BOOST_UP	‘AudioBassBoostUp’
VK_BASS_BOOST_DOWN	‘AudioBassBoostDown’
VK_INFO	‘Info’
VK_GUIDE	‘Guide’
VK_TELETEXT	‘Teletext’
VK_SUBTITLE	‘Subtitle’
VK_BACK	‘BrowserBack’
VK_MENU	‘Menu’
VK_PLAY_PAUSE	‘MediaPlayPause’

NOTE

Numpad values are not distinguished from other numerical key values in this set; a content author could use the KeyboardEvent.location attribute to discover if a key originated from the numeric keypad.

6.3 CSSOM view

Terminals SHALL support `innerWidth` and `innerHeight` properties of the `Window` interface as defined in [CSSOM-VIEW]. Terminals SHOULD support other features defined in [CSSOM-VIEW].

NOTE

Many of the features included in CSSOM-VIEW have been supported in browsers for a long time. For example, the `Screen` interface or extensions to the `Window` interface to obtain the width of the viewport through script. Future versions of this profile may mandate support for this specification, once it reaches a higher level of maturity.

7 Application APIs

A number of specifications from W3C or other organizations define APIs that provide additional functionalities to applications. This section lists which specifications SHALL be supported by a terminal. Note that the APIs defined as part of the HTML5 specification are not included in this section but can be found in annex A.1.3.

Terminals SHALL support the XMLHttpRequest API [XHR-20121206] as profiled in annex A.3.1.

Terminals SHALL support the Web Messaging API [POSTMSG-20120501].

Terminals SHALL support the Web Socket API [WEBSOCKETS-API-20120920] with the protocol defined in [RFC6455].

Terminals SHALL support the Web Workers API [WEBWORKERS-20120501] as profiled in annex A.3.2.

Terminals SHALL support the Server-Sent Events API [SSE-20121211].

Terminals SHALL support the Web Storage API [WEBSTORAGE-20130730].

8 Media Formats and Protocols

8.1 Media Fragment URI

Whenever an aApplication points to a media resource through a URI (e.g. via the `src` attribute of the `<video>` element), it MAY include a Fragment Identifier [RFC3986]. The semantics of such fragments for resources of type audio/*, image/* and video/* are defined in [MEDIA-FRAGS-20120925].

Terminals SHALL support temporal clipping based on Normal Play Time as defined in section 4.2.1 of the Media Fragments URI specification [MEDIA-FRAGS-20120925].

NOTE

There is no requirement to support other means of specifying temporal clipping mentioned in that document such as SMPTE timecodes or real-world clock time or to support the URI query mechanism.

9 Memory Usage (informative)

This section provides some guidelines about the recommended minimum memory requirements of some of the specifications referenced by this document.

Feature	**Memory requirement**	**Comment**
Downloadable fonts [WOFF-20121213]	• 1 MB for Western Fonts. • 10 MB for Asian fonts.	It is recommended to cache WOFF packages for for a better user experience.
Web Storage [WEBSTORAGE-20130730]	• 1 MB per origin. • 10MB global.	1. Terminals may provide methods to free-up memory as described in [WEBSTORAGE-20130730] 2. If the global limit is exceeded, a Terminal may not be able to guarantee the per-origin storage limit
Cookies [COOKIES]	• At least 4 096 bytes per cookie (as measured by the sum of the length of the cookie's name, value, and attributes). • At least 20 cookies per domain. • At least 100 cookies total. • At least 5 120 bytes for the "Set-Cookie" header.	

Annex A Support Tables

This section includes support tables for various specifications referenced by this profile. Each feature is given a short description and hyperlinked to its place within the related specification.

Terminals SHALL support all features marked as "Yes" in the "Mandatory" column; if restrictions apply, the feature support is indicated as "Partial" and the "Comment" column provides a description of the normative requirements. Support for all features marked as "No" is OPTIONAL.

Note that specifications that are required to be fully supported don't have a related support table in this section.

A.1 HTML5 Profile

A.1.1 Elements

This section lists HTML elements that are either new HTML5 elements or elements already defined in HTML4 and now re-defined in HTML5. Some of the HTML4 elements have slightly modified meanings in HTML5 to better reflect how they are used on the Web or to make them more useful.

Element name	Description	Mandatory	Comment
<body>	The `body` element of a document is the first child of the `html` element that is either a `body` element or a `frameset` element. If there is no such element, it is null.	Yes	
<base>	The `base` element allows authors to specify the document base URL for the purposes of resolving relative URLs, and the name of the default browsing context for the purposes of following hyperlinks. The element does not represent any content beyond this information.	Yes	
<meta>	The meta element represents various kinds of metadata that cannot be expressed using the `title`, `base`, `link`, `style`, and `script` elements.	Yes	
<style>	The `style` element allows authors to embed style information in their documents.	Partial	Support for the `scoped` attribute is OPTIONAL.
<br>	The `br` element represents a line break.	Yes	
<div>	The `div` element has no special meaning at all. It represents its children. It can be used with the `class`, `lang`, and `title` attributes to mark up semantics common to a group of consecutive elements.	Yes	
<h1> <h2> <h3> <h4> <h5> <h6>	The div element has no special meaning at all. It represents its children. It can be used with the class, lang, and title attributes to mark up semantics common to a group of consecutive elements.	Yes	
<html>	The `html` element represents the root of an HTML document.	Yes	

Element name	Description	Mandatory	Comment
`<iframe>`	The `iframe` element represents a nested browsing context.	Partial	Support for the `sandbox`, `srcdoc` and `seamless` attributes is OPTIONAL.
`<img>`	An `img` element represents an image.	Yes	
`<li>`	The `li` element represents a list item.	Yes	
`<link>`	The `link` element allows authors to link their document to other resources.	Partial	1. Support for the `sizes` attribute is OPTIONAL. 2. Support for the IDL attribute `relList` is OPTIONAL
`<ol>`	The `ol` element represents a list of items, where the items have been intentionally ordered, such that changing the order would change the meaning of the document.	Yes	
`<option>`	The `option` element represents an option in a select element or as part of a list of suggestions in a datalist element.	Yes	
`<p>`	The `p` element represents a paragraph.	Yes	
`<pre>`	The `pre` element represents a block of preformatted text, in which structure is represented by typographic conventions rather than by elements.	Yes	
`<dl>`	The `dl` element represents an association list consisting of zero or more name-value groups (a description list).	Yes	
`<dt>`	The `dt` element represents the term, or name, part of a term-description group in a description list (`dl` element).	Yes	
`<dd>`	The `dd` element represents the description, definition, or value, part of a term-description group in a description list (`dl` element).	Yes	
`<em>`	The `em` element represents stress emphasis of its contents.	Yes	
`<s>`	The `s` element represents contents that are no longer accurate or no longer relevant.	Yes	
`<q>`	The `q` element represents some phrasing content quoted from another source.	Yes	
`<dfn>`	The `dfn` element represents the defining instance of a term.	Yes	
`<abbr>`	The `abbr` element represents an abbreviation or acronym, optionally with its expansion.	Yes	
`<code>`	The `code` element represents a fragment of computer code.	Yes	

Element name	Description	Mandatory	Comment
`<samp>`	The `samp` element represents (sample) output from a program or computing system.	Yes	
`<var>`	The `var` element represents a variable	Yes	
`<kbd>`	The `kbd` element represents user input (typically keyboard input, although it may also be used to represent other input, such as voice commands).	Yes	
`<sup>,<sub>`	The `sup` element represents a superscript and the `sub` element represents a subscript.	Yes	
`<u>`	The `u` element represents a span of text with an unarticulated, though explicitly rendered, non-textual annotation, such as labeling the text as being a proper name in Chinese text (a Chinese proper name mark), or labeling the text as being misspelt.	Yes	
`<bdi>`	The `bdi` element represents a span of text that is to be isolated from its surroundings for the purposes of bidirectional text formatting.	No	
`<bdo>`	The `bdo` element represents explicit text directionality formatting control for its children. It allows authors to override the Unicode bidirectional algorithm by explicitly specifying a direction override.	Yes	
`<ins>`	The `ins` element represents an addition to the document.	Yes	
`<del>`	The `del` element represents a removal from the document.	Yes	
`<object>`	The `object` element can represent an external resource, which, depending on the type of the resource, will either be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin.	Yes	
`<param>`	The `param` element defines parameters for plugins invoked by object elements. It does not represent anything on its own.	Yes	
`<map>`	The `map` element, in conjunction with any area element descendants, defines an image map. The element represents its children.	Yes	
`<area>`	The `area` element represents either a hyperlink with some text and a corresponding area on an image map, or a dead area on an image map.	Partial	Support for the IDL attribute `relList` is OPTIONAL.
`<caption>`	The `caption` element represents the title of the `table` that is its parent, if it has a parent and that is a `table` element.	Yes	
`<table>`	The `table` element represents data with more than one dimension, in the form of a table.	Partial	Support for the `summary` attribute is OPTIONAL.

Element name	Description	Mandatory	Comment
<colgroup>	The colgroup element represents a group of one or more columns in the table that is its parent, if it has a parent and that is a table element.	Yes	
<col>	If a col element has a parent and that is a colgroup element that itself has a parent that is a table element, then the col element represents one or more columns in the column group represented by that colgroup.	Yes	
<tbody>	The tbody element represents a block of rows that consist of a body of data for the parent table element, if the tbody element has a parent and it is a table.	Yes	
<thead>	The thead element represents the block of rows that consist of the column labels (headers) for the parent table element, if the thead element has a parent and it is a table.	Yes	
<tfoot>	The tfoot element represents the block of rows that consist of the column summaries (footers) for the parent table element, if the tfoot element has a parent and it is a table.	Yes	
<tr>	The tr element represents a row of cells in a table.	Yes	
<td>		Yes	
<th>	The th element represents a header cell in a table.	Yes	
<form>	The form element represents a collection of form-associated elements, some of which can represent editable values that can be submitted to a server for processing.	Yes	
<fieldset>	The fieldset element represents a set of form controls optionally grouped under a common name.	Partial	Support for the name and type attributes is OPTIONAL.
<legend>	The legend element represents a caption for the rest of the contents of the legend element's parent fieldset element, if any.	Yes	
<button>	The button element represents a button.	Yes	
<optgroup>	The optgroup element represents a group of option elements with a common label.	Yes	
<script>	The script element allows authors to include dynamic script and data blocks in their documents. The element does not represent content for the user.	Partial	Support for the async attribute is OPTIONAL.

Element name	Description	Mandatory	Comment
`<noscript>`	The `noscript` element represents nothing if scripting is enabled, and represents its children if scripting is disabled. It is used to present different markup to user agents that support scripting and those that don't support scripting, by affecting how the document is parsed.	No	Since support for scripting is mandated by this profile, support for this element is not needed.
`<span>`	The `span` element doesn't mean anything on its own, but can be useful when used together with the global attributes, e.g. `class`, `lang`, or `dir`. It represents its children.	Yes	
`<title>`	The `title` element represents the document's title or name.	Yes	
`<ul>`	The `ul` element represents a list of items, where the order of the items is not important — that is, where changing the order would not materially change the meaning of the document.	Yes	
`<section>`	Represents a generic document or application section. It can be used together with the h1, h2, h3, h4, h5, and h6 elements to indicate the document structure.	Yes	
`<article>`	Represents an independent piece of content of a document, such as a blog entry or newspaper article.	Yes	
`<aside>`	Represents a piece of content that is only slightly related to the rest of the page.	Yes	
`<hgroup>`	Represents the header of a section.	Yes	
`<header>`	Represents a group of introductory or navigational aids.	Yes	
`<footer>`	Represents a footer for a section and can contain information about the author, copyright information, etc.	Yes	
`<nav>`	Represents a section of the document intended for navigation.	Yes	
`<figure>`	Can be used to associate a caption together with some embedded content, such as a graphic or video:	Yes	
`<figcaption>`	Provides the caption for the `<figure>` element's contents.	Yes	
`<video>`	Represents a video or movie. It is a media element whose media data is ostensibly video data, possibly with associated audio data.	Partial	Support for `crossorigin`, `mediagroup`, and `controls` content attributes is OPTIONAL.
`<audio>`	Represents a sound or audio stream. It is a media element whose media data is ostensibly audio data.	Partial	Support for `crossorigin`, `mediagroup`, and `controls` content attributes is OPTIONAL.

Element name	Description	Mandatory	Comment
`<track>`	The track element allows authors to specify explicit external timed text tracks for media elements. It does not represent anything on its own.	No	Not required as only support for in-band tracks is required.
`<source>`	Allows authors to specify multiple media resources for media elements. It does not represent anything on its own.	Yes	
`<embed>`	It is is used for plug-in content. It represents an integration point for an external (typically non-HTML) applications or interactive content.	Yes	
`<mark>`	Represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.	Yes	
`<progress>`	Represents a completion of a task such as downloading, or when performing a series of expensive operations.	Yes	
`<meter>`	Represents a scalar measurement within a known range, or a fractional value; for example: • disk usage • the relevance of a query result • the fraction of a voting population to have selected a particular candidate	Yes	
`<time>`	Represents either a time on a 24 hour clock, or a precise date in the proleptic Gregorian calendar, optionally with a time and a time zone. It provides an API for accessing the date/time as a Date object.	No	
`<ruby>`	Allows one or more spans of phrasing content to be marked with ruby annotations. Ruby annotations are short runs of text presented alongside base text, primarily used in East Asian typography as a guide for pronunciation or to include other annotations. In Japanese, this form of typography is also known as furigana.	No	
`<rt>`	Marks the ruby text component of a ruby annotation.	No	
`<rp>`	Can be used to provide parentheses around a ruby text component of a ruby annotation, to be shown by user agents that do not support ruby annotations.	No	
`<wbr>`	Represents a line-break opportunity where phrasing content is expected. For example, someone is quoted as saying something which, for effect, is written as one long word. However, to ensure that the text can be wrapped in a readable fashion, the individual words in the quote are separated using a `<wbr>` element.	Yes	

Element name	Description	Mandatory	Comment
`<canvas>`	Represents a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, or other visual images on the fly.	Partial	For details on support of Canvas APIs, see annex A.3.3
`<command>`	Represents a command that the user can invoke.	No	
`<details>`	Represents additional information or controls which the user can obtain on demand.	No	
`<summary>`	Represents a summary, caption, or legend for the rest of the contents of the summary element's parent details element, if any.	No	
`<datalist>`	Represents a set of `<option>` elements that represent predefined options for other controls. The contents of the element represents fallback content for legacy user agents, intermixed with `<option>` elements that represent the predefined options. In the rendering, the `<datalist>` element represents nothing and it, along with its children, should be hidden. The `<datalist>` element together with the new `list` attribute for input can be used to make comboboxes.	Yes	
`<keygen>`	Represents a key pair generator control. When the control's form is submitted, the private key is stored in the local keystore, and the public key is packaged and sent to the server.	Yes	
`<output>`	Represents some type of output, as from a calculation done through scripting.	Yes	
`<textarea>`	Represents a multiline plain text edit control for the element's raw value	Yes	
`<select>`	Represents a control for selecting amongst a set of options.	Yes	

Element name	Description	Mandatory	Comment
`<input>`	The `<input>` element represents a typed data field, usually with a form control to allow the user to edit the data. The `<input>` element `type` attribute has the following values. - `"text"` - `"url"` - `"email"` - `"datetime"` - `"date"` - `"month"` - `"week"` - `"time"` - `"datetime-local"` - `"number"` - `"range"` - `"tel"` - `"search"` - `"color"` - `"checkbox"` - `"image"` - `"file"`	Partial	1. The following values for the `type` attribute SHALL be supported: `"text"`, `"url"`, `"email"`, `"number"`, `"range"`, `"tel"`, `"search"`, `"checkbox"`, `"date"`; all other values are OPTIONAL to support (note that some are marked as feature at risk in [HTML5-20130806]) . 2. Support for the `autocomplete` attribute is OPTIONAL.
`<a>`	The `<a>` element *without* an `href` attribute now represents a "**placeholder link**". It can also contain flow content rather than being restricted to phrase content.	Partial	Support for the IDL attribute `relList` is OPTIONAL.
`<address>`	The `<address>` element is now scoped by the new concept of sectioning. It represents the contact information for its nearest article or body element ancestor. If that is the body element, then the contact information applies to the document as a whole.	Yes	
`<b>`	The `<b>` element represents a span of text to be stylistically offset from the normal prose without conveying any extra importance, such as key words in a document abstract, product names in a review, or other spans of text whose typical typographic presentation is boldened.	Yes	
`<cite>`	The `<cite>` element now solely represents the title of a work (e.g. a book, a paper, an essay, a poem, a score, a song, a script, a film, a TV show, a game, a sculpture, a painting, a theatre production, a play, an opera, a musical, an exhibition, a legal case report, etc). Specifically the example in HTML4 where it is used to mark up the name of a person is no longer considered conforming.	Yes	
`<hr>`	The `<hr>` element represents a paragraph-level thematic break, e.g. a scene change in a story, or a transition to another topic within a section of a reference book.	Yes	

Element name	Description	Mandatory	Comment
`<i>`	The `<i>` element represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, a thought, a ship name, or some other prose whose typical typographic presentation is italicized.	Yes	
`<label>`	For the `<label>` element the browser should no longer move *focus* from the label to the control unless such behavior is standard for the underlying platform user interface.	Yes	
`<menu>`	The `<menu>` element represents a list of commands and is redefined to be useful for toolbars and context menus.	Partial	1. `<menu>` element of `type` *list* SHALL be supported. Support for other types is OPTIONAL. 2. Support for the `label` attribute is OPTIONAL.
`<small>`	The `<small>` element now represents small print (for side comments and legal print).	Yes	
`<strong>`	The `<strong>` element now represents importance rather than strong emphasis.	Yes	
`<head>`	The `<head>` element no longer allows the object element as child.	Yes	

A.1.2 Global attributes

Attribute name	Description	Mandatory	Comment
`accessKey`	The accesskey attribute's value is used by the user agent as a guide for creating a keyboard shortcut that activates or focuses the element.	No	
`class`	Every HTML element may have a class attribute specified. The attribute, if specified, must have a value that is an unordered set of unique space-separated tokens representing the various classes that the element belongs to. The classes that an HTML element has assigned to it consist of all the classes returned when the value of the class attribute is split on spaces.	Yes	
`dir`	The `dir` attribute specifies the element's text directionality (left-to-right, or right-to-left). The attribute is an enumerated attribute with the keyword `ltr` ((left-to-right) mapping to the state `ltr`, and the keyword `rtl` (right-to-left) mapping to the state `rtl`. The attribute has no defaults.	Yes	
`id`	The `id` attribute represents its element's unique identifier. The value must be unique in the element's home subtree and must contain at least one character. The value must not contain any space characters.	Yes	

Attribute name	Description	Mandatory	Comment
`lang`	The `lang` attribute (in no namespace) specifies the primary language for the element's contents and for any of the element's attributes that contain text. Its value must be a valid BCP 47 language code, or the empty string.	Yes	
`style`	All HTML elements may have the `style` content attribute set. If specified, the attribute must contain only a list of zero or more semicolon-separated (;) CSS declarations. In user agents that support CSS, the attribute's value must be parsed when the attribute is added or has its value changed, with its value treated as the body (the part inside the curly brackets) of a declaration block in a rule whose selector matches just the element on which the attribute is set. All URLs in the value must be resolved relative to the element when the attribute is parsed. For the purposes of the CSS cascade, the attribute must be considered to be a `style` attribute at the author level. Documents that use `style` attributes on any of their elements *must still be comprehensible and usable if those attributes were removed*.	Yes	
`tabindex`	The `tabindex` content attribute specifies : • If the element is focusable • If it can be reached using sequential focus navigation • The relative order of the element for the purposes of sequential focus navigation The name *"tab index"* comes from the common use of the `tab` key to navigate through the focusable elements. The term "tabbing" refers to moving forward through the focusable elements that can be reached using sequential focus navigation. If it is specified, it must have a value that is a valid integer, and it must be parsed using the rules for parsing integers.	Yes	
`title`	The `title` attribute represents advisory information: • For an element, such as would be appropriate for a tooltip • On a link, this could be the title or a description of the target resource • On an image, it could be the image credit or a description of the image • On a paragraph, it could be a footnote or commentary on the text • On a citation, it could be further information about the source • etc. The value is text.	Yes	

Attribute name	Description	Mandatory	Comment
contenteditable / isContentEditable	The contenteditable attribute is an enumerated attribute whose keywords are the empty string, true, and false. The empty string and the true keyword map to the true state. The false keyword maps to the false state. In addition, there is a third state, the inherit state, which is the missing value default (and the invalid value default). • The true state indicates that the element is editable. • The inherit state indicates that the element is editable if its parent is. • The false state indicates that the element is not editable.	No	
designMode	Documents have a designMode, which can be either enabled or disabled. When enabled, the document is editable.	No	
contextmenu	The contextmenu attribute gives the element's context menu. The value must be the ID of a menu element in the DOM. If the node that would be obtained by the invoking the getElementById() method using the attribute's value as the only argument is null or not a menu element, then the element has no assigned context menu. Otherwise, the element's assigned context menu is the element so identified.	No	
data-*	A custom data attribute is an attribute in no namespace whose name starts with the string "data-", has at least one character after the hyphen, is XML-compatible, and contains no characters in the range U+0041 .. U+005A (LATIN CAPITAL LETTER A .. LATIN CAPITAL LETTER Z). Custom data attributes are intended to store custom data private to the page or application, for which there are no more appropriate attributes or elements. These attributes are not intended for use by software that is independent of the site that uses the attributes. Every HTML element may have any number of custom data attributes specified, with any value. They refer to the DOM Dataset API.	Yes	
draggable	All HTML elements may have the draggable content attribute set. It is an enumerated attribute, and has three states: • The first state is true and it has the keyword true. The true state means the element is draggable. • The second state is false and it has the keyword false. The false state means that it is not draggable. • The third state is auto; it has no keywords but it is the missing value default. The auto state uses the default behavior of the user agent.	No	
dropzone	All HTML elements may have the dropzone content attribute set.	No	

Attribute name	Description	Mandatory	Comment
hidden	All HTML elements may have the `hidden` content attribute set. The hidden attribute is a *boolean* attribute. When specified on an element, it indicates that the element is not yet, or is no longer, relevant. User agents should not render elements that have the hidden attribute specified.	Yes	
spellcheck	The `spellcheck` attribute is an enumerated attribute whose keywords are the `empty string`, `true` and `false`. The empty string and the `true` keyword map to the true state. The `false` keyword maps to the false state. Please note that the "default" state has been removed. • The true state indicates that the element is to have its spelling and grammar checked. • The false state indicates that the element is not to be checked.	No	

A.1.3 Web applications APIs

HTML5 introduces a number of APIs that help creating applications.

API name	Description	Mandatory	Comment
Media elements	An API for playing of video and audio which can be used with the new `<video>` and `<audio>` elements.	Partial	See profile in annex A.1.3.1
Application cache	An API that enables offline Web applications.	No	This feature is potentially useful but marked as feature at risk in [HTML5-20130806] and likely to change in the near future.
Custom scheme and content handlers	An API that allows a Web application to register itself for certain protocols or media types.	No	Not relevant for this profile. Also, marked as feature at risk in [HTML5-20130806]
Custom search provider	An API that allows a Web application to register itself for certain search provider.	No	Not relevant for this profile. Also, marked as feature at risk in [HTML5-20130806]
Drag and drop	This API works in combination with a `draggable` attribute.	No	
History	An API that exposes the history and allows pages to add to it to prevent breaking the back button.	Yes	
Base64 utility methods	An API that allow authors to transform content to and from the base64 encoding.	Yes	

A.1.3.1 Media APIs

HTML Media APIs SHALL be supported as profiled below

Interface	Attribute Name	Mandatory	Comment
HTMLVideoElement	width	Yes	
	height	Yes	
	videoWidth	Yes	
	videoHeight	Yes	
	poster	Yes	
HTMLAudioElement		Yes	
HTMLSourceElement	src	Yes	
	type	Yes	
	media	Yes	
HTMLTrackElement		No	
HTMLMediaElement	error	Yes	
	src	Yes	
	currentSrc	Yes	
	crossOrigin	No	
	networkState	Yes	
	preload	Yes	
	buffered	Yes	
	load()	Yes	
	canPlayType()	Yes	
	readyState	Yes	
	seeking	Yes	
	currentTime	Yes	
	duration	Yes	
	startDate	No	
	paused	Yes	
	defaultPlaybackRate	Yes	
	playbackRate	Yes	
	played	Yes	
	seekable	Yes	
	ended	Yes	
	autoplay	Yes	

	loop	Yes	
	play()	Yes	
	pause()	Yes	
	mediaGroup	No	
	controller	No	
	controls	No	
	volume	Yes	
	muted	Yes	
	defaultMuted	Yes	
	videoTracks	No	
	audioTracks	Yes	
	textTracks	Yes	Only support for in-band tracks is required
	addTextTrack()	No	
MediaError	code	Yes	
AudioTrackList	length	Yes	
	getter(index)	Yes	
	getTrackByID()	Yes	
	onchange	Yes	
	onaddtrack	No	
	onremovetrack	No	
AudioTrack	id	Yes	
	kind	Yes	
	label	Yes	
	language	Yes	
	enabled	Yes	
VideoTrackList		No	
VideoTrack		No	
TextTrackList		Yes	
TextTrack	kind	Yes	
	label	Yes	
	language	Yes	
	onMetadataTrackDispatchType	Yes	

	mode	Yes	
	cues	No	
	activeCues	No	
	addCue	No	
	removeCue	No	
	oncuechange	No	
TextTrackCueList		No	
TextTrackCue		No	
MediaController		No	
TimeRange	length	Yes	
	start	Yes	
	end	Yes	
TrackEvent		Yes	

A.1.3.2 Media Element Events Support

Events	Mandatory
loadstart	Yes
progress	Yes
suspend	Yes
abort	Yes
error	Yes
emptied	Yes
stalled	Yes
loadedmetadata	Yes
loadeddata	Yes
canplay	Yes
canplaythrough	Yes
playing	Yes
waiting	Yes
seeking	Yes
seeked	Yes
ended	Yes
durationchange	Yes

`timeupdate`	Yes
`play`	Yes
`pause`	Yes
`ratechange`	Yes
`volumechange`	Yes

A.2 CSS3 profile

This section contains support tables for various CSS specifications.

A.2.1 CSS Basic User Interface

A.2.1.1 Properties and Values

Property	Values	Description	Mandatory	Comment
`appearance`	normal | <appearance> | inherit	This property can be used to make an element look like a standard user interface element on the platform. It is a shorthand for "appearance", "color", "font", and "cursor".	No	
`box-sizing`	padding-box | content-box | border-box | inherit	This property specifies a border/padding value in relation to a fluid length element.	Partial	Support for property value `padding-box` is OPTIONAL (marked as feature at risk in [HTML5-20130806])
`content`	icon	This property is used with the :before and :after pseudo-elements to generate content in a document.	No	Marked as feature at risk in [HTML5-20130806]
`cursor`	[[<uri> [<x> <y>]?,]* [auto | default | none | context-menu | help | pointer | progress | wait | cell | crosshair | text | vertical-text | alias | copy | move | no-drop | not-allowed | e-resize | n-resize | ne-resize | nw-resize | s-resize | se-resize | sw-resize | w-resize | ew-resize | ns-resize | nesw-resize | nwse-resize | col-resize | row-resize | all-scroll | zoom-in | zoom-out]] | inherit	This property specifies the type of cursor to be displayed for the pointing device.	Yes	
`font`	<appearance> | status-bar | message-box | caption | small-	The shorthand "font" property has the effect of setting all of the	Yes	

	caption | inherit	elemental "font-*" properties.		
icon	auto | <uri> [, <uri>]* | inherit	The "icon" property give the author the ability to style any arbitrary element with an iconic equivalent.	No	Feature at risk in [HTML5-20130806]
nav-index	auto | <number> | inherit	This property is an input-method-neutral way of specifying the sequential navigation order, also known as "tabbing order".	Yes	
nav-up	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-up focus.	Yes	
nav-right	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-right focus.	Yes	
nav-down	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-down focus.	Yes	
nav-left	auto | <id> [current | root | <target-name>]? | inherit	This property allows User agents for devices with directional navigation keys to respond by navigating the nav-left focus.	Yes	
outline	[<"outline-color"> || <"outline-style"> || <"outline-width">] | inherit	This property allows authors who want to create outlines around visual objects such as buttons, active form fields, image maps, etc., to make them stand out.	Yes	
outline-color	<color> | invert | inherit	This property controls the border color of a dynamic outline.	Yes	
outline-offset	<length> | inherit	This property controls the border offset of a	Yes	

		dynamic outline.		
outline-style	auto | <border-style> | inherit	This property controls the border style of a dynamic outline.	Partial	At this time `outline-style` SHALL be supported as defined by CSS2.1 specification, so support for value `auto` is not required.
outline-width	<border-width> | inherit	This property controls the border width of a dynamic outline.	Yes	
resize	none | both | horizontal | vertical | inherit	This property allows an author to specify whether or not an element is resizable by the user, and if so, along which axis/axes.	No	
ime-mode	auto | normal | active | inactive | disabled | inherit	The ‘ime-mode’ CSS property controls the state of the input method editor for text fields.	No	Feature at risk in [HTML5-20130806]
text-overflow	(clip | ellipsis |){1,2} | inherit	This property specifies rendering when inline content overflows its block container element ("the block") in its inline progression direction that has ‘overflow’ other than ‘visible’	Partial	Support for the `text-overflow` property value <string> and the `text-overflow` property 2-value syntax and definition is OPTIONAL (marked as feature at risk in [CSS3UI-20120117]).

A.2.1.2 User interface pseudo classes

Pseudo class	Mandatory	Comment
:active	Yes	
:checked	Yes	
:default	Yes	
:disabled	Yes	
:enabled	Yes	
:focus	Yes	
:hover	Yes	
:in-range	Yes	
:indeterminate	Yes	

`:invalid`	Yes	
`:optional`	Yes	
`:out-of-range`	Yes	
`:read-only`	Yes	
`:read-write`	Yes	
`:required`	Yes	
`:valid`	Yes	
`:visited`	Yes	

A.2.2 CSS Image Values and Replaced Content

Terminals SHALL support Gradients as defined in CSS Image Values and Replaced Content [CSS3-IMAGES-20120417]. Support for other features is OPTIONAL.

A.2.3 CSS Backgrounds and Borders

Terminals SHALL support all features in CSS Backgrounds and Borders [CSS3-BG-20120724] with the exception of the `border-image` set of properties.

A.2.4 CSS Fonts Module Level 3

Terminals SHALL support the [CSS3-FONTS-20130212] specification as profiled below:

1. The `font-family`, `font-weight`, `font-style`, `font-size` and `font` properties SHALL be supported. Support for other properties is OPTIONAL.

2. Support for the values **caption**, **icon**, **menu**, **message-box**, **small-caption**, **status-bar** of the `font` property is OPTIONAL.

3. The `@font-face` rule SHALL be supported. Support for other rules is OPTIONAL.

4. **src**, **font-family**, **font-size** and **font-weight** descriptors for the `@font-face` rule SHALL be supported. Support for other descriptors is OPTIONAL.

A.2.5 CSS Text Level 3

Terminals SHALL support `text-shadow`.

A.2.6 CSS Transforms

1. Only support for the two-dimensional subset of the specification is required. See section "Two Dimensional Subset" of [CSS3-TRANSFORMS-20120911]

   NOTE

   This implies that
   - only support for `transform` and `transform-origin` properties is required
   - only support for `matrix()`, `translate()`, `translateX()`, `translateY()`, `scale()`, `scaleX()`, `scaleY()`, `rotate()`, `skewX()`, `skewY()` transform functions is required.

2. Transform function lists SHALL be supported

3. Support for the SVG extensions is not required

A.2.7 CSS Transitions

Support for the `pseudoElement` field on the `TransitionEvent` interface is OPTIONAL.

Support for the `pseudoElementArg` argument for the `initTransitionEvent` method is OPTIONAL

The list of “animatable” properties in [CSS3-TRANSITIONS-20130212] SHALL be profiled as indicated in the table below. For any other property not listed below, the following holds: if a property is supported by this profile and is defined to be "animatable" by the related CSS spec, than it SHALL be animatable, unless explicitly excluded by this specification.

Property Name	Type	Madatory
`background-color`	color	Yes
`background-image`	only gradients	No
`background-position`	percentage, length	No
`border-bottom-color`	color	No
`border-bottom-width`	length	Yes
`border-color`	color	No
`border-left-color`	color	No
`border-left-width`	length	Yes
`border-right-color`	color	No
`border-right-width`	length	Yes
`border-spacing`	length	Yes
`border-top-color`	color	No
`border-top-width`	length	Yes
`border-width`	length	Yes
`bottom`	length, percentage	Yes
`color`	color	Yes
`crop`	rectangle	No
`font-size`	length, percentage	Yes
`font-weight`	number	Yes
`grid-*`	various	No
`height`	length, percentage	Yes
`left`	length, percentage	Yes
`letter-spacing`	length	Yes
`line-height`	number, length, percentage	Yes
`margin-bottom`	length	Yes

margin-left	length	Yes
margin-right	length	Yes
margin-top	length	Yes
max-height	length, percentage	Yes
max-width	length, percentage	Yes
min-height	length, percentage	Yes
min-width	length, percentage	Yes
opacity	number	Yes
outline-color	color	Yes
outline-offset	integer	Yes
outline-width	length	Yes
padding-bottom	length	Yes
padding-left	length	Yes
padding-right	length	Yes
padding-top	length	Yes
right	length, percentage	Yes
text-indent	length, percentage	Yes
text-shadow	shadow	No
top	length, percentage	Yes
vertical-align	keywords, length, percentage	Yes
visibility	visibility	Yes
width	length, percentage	Yes
word-spacing	length, percentage	Yes
z-index	integer	Yes
zoom	number	No

A.3 Web APIs profile

A.3.1 XMLHttpRequest

Support for `Document` response type is OPTIONAL. All other features SHALL be supported.

A.3.2 Web Workers

Support for Shared Workers is OPTIONAL

A.3.3 Canvas 2D

Support for the following interfaces is not required by this profile

- `Path`
- `DrawingStyle`
- `HitRegionOptions`

All the others SHALL be supported as profiled below

A.3.3.1 HTMLCanvasElement

Member	Support	Comment
`width`	Yes	
`height`	Yes	
`toDataURL()`	Yes	
`getContext(context)`	Yes	

A.3.3.2 TextMetrics

Member	Mandatory	Comment
`width`	Yes	
`actualBoundingBoxLeft`	No	
`actualBoundingBoxRight`	No	
`fontBoundingBoxAscent`	No	
`fontBoundingBoxDescent`	No	
`actualBoundingBoxAscent`	No	
`actualBoundingBoxDescent`	No	
`emHeightAscent`	No	
`emHeightDescent`	No	
`hangingBaseline`	No	
`alphabeticBaseline`	No	
`ideographicBaseline`	No	

A.3.3.3 CanvasGradient

Member	Mandatory	Comment
`addColorStop(offset, color)`	Yes	

A.3.3.4 CanvasRenderingContext2D

Member	Mandatory	Comment
`canvas`	Yes	
`save()`	Yes	
`restore()`	Yes	
`scale(x, y)`	Yes	
`rotate(angle)`	Yes	
`translate(x, y)`	Yes	
`transform(m11, m12, m21, m22, dx, dy)`	No	
`setTransform(m11, m12, m21, m22, dx, dy)`	No	
`globalAlpha`	Yes	
`globalCompositeOperation`	Yes	The value `source-over` for this property SHALL be supported. Support for other values is OPTIONAL.
`strokeStyle`	Yes	
`fillStyle`	Yes	
`createImageData(sw, sh)`	Yes	
`createImageData(imagedata)`	Yes	
`createLinearGradient(x0, y0, x1, y1)`	Yes	If this feature is not hardware accelerated, performance will be limited
`createRadialGradient(x0, y0, r0, x1, y1, r1)`	Yes	If this feature is not hardware accelerated, performance will be limited
`createPattern(image, repetition)`	Yes	
`lineWidth`	Yes	
`lineCap`	Yes	If this feature is not hardware accelerated, performance will be limited
`lineJoin`	Yes	
`miterLimit`	Yes	
`shadowOffsetX`	Yes	
`shadowOffsetY`	Yes	
`shadowBlur`	Yes	If this feature is not hardware accelerated, performance will be limited
`shadowColor`	Yes	
`clearRect(x, y, w, h)`	Yes	

Member	Mandatory	Comment
`fillRect(x, y, w, h)`	Yes	
`strokeRect(x, y, w, h)`	Yes	If this feature is not hardware accelerated, performance will be limited
`beginPath()`	Yes	
`closePath()`	Yes	
`moveTo(x, y)`	Yes	
`lineTo(x, y)`	Yes	
`quadraticCurveTo(cpx, cpy, x, y)`	No	
`bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)`	No	
`arcTo(x1, y1, x2, y2, radius)`	No	
`rect(x, y, w, h)`	Yes	
`arc(x, y, radius, startAngle, endAngle, anticlockwise)`	No	
`ellipse(x, y, radiusX, radiusY, rotation, startAngle, endAngle, anticlockwise);`	No	
`fill()`	Yes	
`fill(path)`	No	
`stroke()`	Yes	
`stroke(path)`	No	
`clip()`	No	
`clip(path)`	No	
`isPointInPath(x, y)`	Yes	
`isPointInPath(path, x, y)`	No	
`drawImage(image, dx, dy)`	Yes	
`drawImage(image, dx, dy, dw, dh)`	Yes	
`drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh)`	Yes	
`getImageData(sx, sy, sw, sh)`	Yes	
`putImageData(image, dx, dy)`	Yes	
`putImageData(image, dx, dy, dirtyX, dirtyY, dirtyWidth, dirtyHeight)`	No	
`font`	Yes	
`textAlign`	Yes	
`textBaseline`	Yes	

Member	Mandatory	Comment
`fillText`	Yes	
`strokeText`	Yes	
`measureText`	Yes	
`setLineDash(segments)`	No	
`getLineDash()`	No	
`lineDashOffset`	No	
`drawSystemFocusRing(element)`	No	
`drawSystemFocusRing(path,element)`	No	
`drawCustomFocusRing(element)`	No	
`drawCustomFocusRing(path,element)`	No	
`scrollPathIntoView`	No	
`scrollPathIntoView(path)`	No	
`addHitRegion(options)`	No	
`removeHitRegion(options)`	No	